DEVICE PRESENTING INFORMATION ABOUT RESOURCE LOCATION OF DEVICE CONTROL SOFTWARE

BACKGROUND OF THE INVENTION

5   1.   Field of the Invention

[0001]   This invention relates to a device used in connection with a network, in particular, to a technique to acquire and update device control software for controlling the device via the network.

10   2.   Description of the Related Art

[0002]   In the prior art, a user of a devices such as printers and routers, which are used in connection with a network, downloads a latest version of device driver, firmware or utility software (hereinafter, software for control devices including these software collectively referred to as "device control software") from a homepage of the device manufacturer.   On the homepage, storage locations of device control software are shown based on a model name of device or an operating system (hereinafter referred to as "OS") to be applied.

[0003]   However, the homepage of manufacturer may have a complicated link structure, and thus it may be difficult to find out the Web page in which the device control software can be downloaded.   Furthermore, since the user must identify the OS used by the user and the model name of the device in order to download, it may be difficult for a beginner to identify the device control software to be downloaded.

25

SUMMARY OF THE INVENTION

[0004]   The invention is made in view of these problems and addresses to enable the users to readily download device control software.

[0005]   A device of the invention is exemplified below to solve at least part of the above problems.   A first device of the invention, which is

used in connection with a network, includes: a communication unit configured to communicate with a client via the network; an identification unit configured to input OS information representing an operating system installed in the client to identify a type of the operating system used in the

5   client; a search unit configured to search a storage location on the network of device control software, which corresponds to the identified type of the operating system and controls the device, from a database in which specifications of operating systems and storage locations of device control software are recorded in association with each other; and an information

10 unit configured to inform the client of the searched storage location of the device control software.

[0006] The device used in connection with the network includes as follows: image input-output devices such as printers, network displays, scanners and digital cameras; and network communication devices such as

15 routers, hubs, modems and access points. The network may be wired or wireless. In order to use any of such devices through the client, the device control software for controlling the device must be installed in the client or the device itself. In most cases, the latest device control software is uploaded in the server on The Internet which is managed by the

20 manufacturer of the device.

[0007] The device of the invention automatically identifies the OS used by the client through the above-mentioned identification unit, and then informs the client of the storage location of the device control software corresponding to the OS. Therefore, the user can readily download the

25 device control software even if the user does not identify the OS installed in the client or does not know the storage location of the device control software.

[0008] In the device of the invention, the communication unit may communicate based on HTTP, and the information unit may generate and

30 send to the client a markup language file including a link to the storage

location of the device control software.

[0009] This configuration enables the user to readily acquire the device control software only by clicking the link included in the markup language file. Where the markup language includes languages such as SGML, XML, XHTML and HTML.

[0010] In the device of the invention, the communication unit may communicate based on HTTP, and the OS information may be described as environment variable in a HTTP request sent from the client.

[0011] When the device is accessed base on HTTP via the client, a Web browser installed in the client is typically used. Since the Web browser adds the information of the client's OS to the request on sending the request to the device, the device can use this information to identify the type of the OS.

[0012] In the device of the invention, it is possible to provide the database, which associates the types of the operating systems to the storage locations of the device control software, in a variety of locations. For example, it may be stored in the device or in a predetermined server connected to the network. In the latter case, it is possible to readily respond to the change of the storage location of the device control software by rewriting the database on the server. The database is preferably provided in the server managed by the manufacturer of the device since most of changes of storage location are performed by the manufacturer of the device.

[0013] In the case where the database is stored in the server, preferably, the storage locations in the database are also recorded in association with model information of devices and the search unit uses the model information of the device and the identified type of the operating system to search the storage location of the device control software.

[0014] The above-mentioned database can be described in XML to improve its versatility.

[0015] The invention may be configured as the device described

3

below. A second device of the invention, which is used in connection with a network, includes: a storage unit configured to store an URL for download corresponding to a Web page which provides device control software to control the device on The Internet; a HTTP communication unit configured

5 to generate a markup language file including a link to the URL for download and sending the markup language file back to a client through a HTTP response in response to a HTTP request from the client.

[0016] According to the second device, the user can send the HTTP request to the device itself to immediately know the URL for download.

10 [0017] In the second device, the storage unit may store an URL for update corresponding to a Web page which provides update information to update the URL for download, and the device may further include an update unit configured to acquire the update information from the Web page based on the URL for update and using the update information to update the URL

15 for download.

[0018] This configuration enables the update of URL for download to be readily responded to.

[0019] The invention is not limited to the aspects of device described above and is possible to be configured in a variety of aspects. For

20 example, it may be configured as a method for informing the client of the storage location on the network of device control software or as a computer program for performing such information. Such computer program may be recorded in a computer-readable recording medium. Such recording medium may include a variety of media such as flexible disk, CD-ROM, DVD,

25 magneto-optics disk, IC card, hard disk and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Fig. 1 is a schematic of a network configuration;

[0021] Fig. 2 is a schematic that shows functional blocks of a

30 program stored in a ROM;

[0022] Fig. 3 is a schematic that shows one example of request;

[0023] Fig. 4 is a schematic that shows one example of HTML page sent to a browser;

[0024] Fig. 5 is a flowchart that shows a process for informing a user of a driver location;

[0025] Fig. 6 is a schematic that shows a driver location database 12 stored in a server SV;

[0026] Fig. 7 is a flowchart that shows a process for informing a user of a driver location in the second embodiment;

[0027] Fig. 8 is a block diagram of a printer PT and the server SV in the third embodiment; and

[0028] Fig. 9 is a flowchart that shows a process for informing a user of a driver location in the third embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0029] Embodiments of the invention are described in the following sequence.

A. General Description of Network

B. First Embodiment

(B1) Configuration of Printer

(B2) Process for Informing of Driver Location

C. Second Embodiment

(C1) Process for Informing of Driver Location

D. Third Embodiment

(D1) Configuration of Printer

(D2) Process for Informing of Driver Location

A. General Description of Network

[0030] Fig. 1 is a schematic of a network configuration that is used in a following embodiment. A client PC is a general-purpose personal

computer in which an OS and a browser are installed. The browser is used to browse Web pages that are provided by a printer PT or a server SV. Although the only one client PC is shown, there may be a plurality of clients.

[0031] The printer PT is connected to the client PC via a LAN. The printer PT has a function of receiving a print job from the client PC and then printing, and a function as Web server. This Web server function enables the printer PT to present status information and setting information of the print job to the client PC. In this embodiment, an IP address of "192.168.0.2" is assumed to be assigned to the printer PT.

[0032] The LAN is connected to The Internet via a router RT. The router RT serves as firewall to prevent a device on The Internet from performing an unauthorized access to the LAN. The server SV is connected to The Internet. The server SV is managed by the manufacturer of the printer PT and supplies latest printer drivers. In this embodiment, an URL of the server SV is assumed to be "www.***.co.jp."

B. First Embodiment:
(B1) Configuration of Printer:
[0033] The printer PT includes a CPU, a RAM and a ROM, and the ROM stores the programs for performing the function as a Web server and functions described below. The ROM also stores model information of the printer PT. The model information means a model name of the printer PT, for example, "LP-1000" or "LP-2000."

[0034] Fig. 2 is a schematic that shows functional blocks of the program stored in the ROM. A HTTP communication module 10 communicates with the client PC based on HTTP (Hyper Text Transport Protocol). A HTTP communication includes messages as follows: "request" for requesting transfer of document or execution of program; and "response" corresponding to the request. For example, if a request that requests transfer of a file "index.html" is sent to the printer PT from the browser, then

the printer PT sends back the specified file as response. Furthermore, when an execution request is sent to the printer PT from the browser, the printer PT informs the browser of an execution result as response.

[0035] The browser installed in the client PC attaches a variety of environment variables to the request and then sends it. Fig. 3 is a schematic that shows one example of request. Fig. 3 shows the request to request execution of a program "driver.cgi" stored in a root directory of a server which host name is "www.***.co.jp." The name of the OS and the browser that are used by the client PC are described in the environment variable shown as "User-Agent" in the request. Fig. 3 shows that the request has been sent by the browser "Browser1" running on the OS "W_98." An OS identification module 11 parses this environment variable to identify the type of the OS used by the client PC.

[0036] Recorded in a driver location database 12 are locations of printer drivers according to types of OS's. As shown in Fig. 2, for example, the printer driver corresponding to the OS "W_98" is stored in the URL "www.***.co.jp/driver/w98.zip." This database can be configured in the form of database described in XML, relational database or CSV format, for example. The database described in XML is relatively suitable for application on The Internet.

[0037] A driver location search module 13 searches the location of the printer driver corresponding to the OS identified by the above-mentioned OS identification module 11 from the driver location database 12.

[0038] A HTML generation module 14 generates a HTML page including the link to the searched location. The generated page is sent to the client as HTTP response. Fig. 4 is a schematic that shows one example of HTML page sent to the browser. The example shown in Fig. 4 displays the type of the OS (W_98) identified by the OS identification module 11 and the link to the URL (www.***.co.jp/driver/w98.zip) where the printer driver is stored. The user can click this link through a mouse operation to readily

download the printer driver corresponding to the OS of the client PC.

(B2) Process for Informing of Driver Location:

[0039] Fig. 5 is a flowchart that shows a process for informing a

5   user of a driver location.

[0040] First, the user inputs the URL corresponding to the printer PT into an address bar AD of the browser (see Fig. 4). In the example of Fig. 4, "192.168.0.2/driver.cgi" is input. The browser sends a request to the printer PT based on this URL (step S10). Once receiving this request, the

10 printer PT identifies the OS of the client PC from the environment variable attached to the request (step S11).

[0041] The driver location search module 13 searches the location of the driver corresponding to the identified OS from the driver location database 12 (step S12). Next, the HTML including the link to the searched

15 location is generated (step S13) and then it is sent to the browser as HTTP response (step S14).

[0042] The above process enables the printer PT to inform the user of the location of the printer driver corresponding to the OS of the client PC.

20 C. Second Embodiment:

[0043] The driver location database 12 may be stored in the server SV, for example. Fig. 6 is a schematic that shows the driver location database 12 stored in the server SV. As shown in Fig. 6, this driver location database 12 stores addresses of locations of printer drivers according to

25 model names of printer and types of OS.

(C1) Process for Informing of Driver Location:

[0044] Fig. 7 is a flowchart that shows a process for informing a user of a driver location in this embodiment.

30 [0045] The user inputs the URL corresponding to the printer PT

8

into the address bar AD of the browser (see Fig. 4). The browser sends the request to the printer PT based on this URL (step S20). Once receiving this request, the printer PT identifies the type of the OS used by the client PC from the environment variable attached to the request (step S21).

[0046] Then, the printer PT requests the transmission of the driver location database 12 to the server SV (step S22). In response to this request, the server SV sends the driver location database 12 to the printer PT (step S23). Such communication preferably uses HTTP. Because HTTP enables to readily communicate over the firewall by the router RT.

[0047] Next, the printer PT searches the location of the printer driver corresponding to the identified OS and the model information of the printer PT from the acquired driver location database 12 (step S24). Subsequently, the HTML including the link to the searched location is generated (step S25) and then it is sent to the client PC as HTTP response (step S26).

[0048] The second embodiment described above enables the manufacturer of the printer to update the database accordingly since the database is stored in the server SV. Therefore, storage locations of drivers can be changed flexibly.

D. Third Embodiment:

(D1) Configuration of Printer:

[0049] The location of the printer driver may be searched on the server SV side. Fig. 8 is a block diagram of the printer PT and the server SV in this embodiment.

[0050] The printer PT includes a HTTP communication module 20, an OS identification module 21, a driver location search module 23, and a HTML generation module 24. In this embodiment, the driver location search module 23 sends the type information of the OS identified by the OS identification module 21 and the model information stored in the ROM of the

printer PT to the server SV, and thus causes the server SV to search the location of the driver.

[0051] The server SV includes a HTTP communication module 30, a database query module 31, and a driver location database 32. The database query module 31 searches the location of the printer driver from the driver location database 32 based on the type information of the OS and the model information of the printer PT that are received by the HTTP communication module 30. The HTTP communication module 30 informs the printer PT of the search result. Recorded in the driver location database 32 are data equal to that shown in Fig. 6.

(D2) Process for Informing of Driver Location:

[0052] Fig. 9 is a flowchart that shows a process for informing a user of a driver location in this embodiment.

[0053] First, the user inputs the URL corresponding to the printer PT into an address bar AD of the browser (see Fig. 4). The browser sends a request to the printer PT based on this URL (step S30). Once receiving this request, the printer PT identifies the OS of the client PC from the environment variable attached to the request (step S31).

[0054] Next, the driver search module specifies the model information of the printer PT read from the ROM and the OS name identified in the previous step as arguments to issue the request to the server SV (step S32). The server SV searches the location of the driver based on these arguments (step S33) and sends the result as response to the printer PT (step S34). The driver location search module delivers the received address of the driver location to the HTML generation module, and the HTML generation module generates the Web page (step S35) and informs the browser of the Web page as response (step S36).

[0055] The third embodiment described above enables the load on the printer PT to be reduced since the driver is searched on the server side.

[0056] Although the embodiments of the invention have been described, it should be noted that the invention is not limited to the above embodiments and may implement various aspects without departing from the spirit of the invention. For example, a matter realized in the form of software may be realized in the form of hardware. Furthermore, although the invention is applied to the printer in the embodiments, it may be applied to various devices such as routers , hubs and ADSL modems, which have a Web server function.

[0057] In the various embodiments described above, the location of the driver corresponding the OS of the client is searched from the database and then informed. Instead, the printer may previously store the URL of the Web page that provides the drivers, and thus the client may be informed of this URL. The Web page that provides drivers, for example, includes a Web page of the manufacturer for providing the drivers, a Web page where the drivers corresponding to the respective OS's are listed. Such aspect also enables the complication of the user searching the Web page for providing the drivers to be reduced. Meanwhile, if the URL stored in the printer can be updated, preferably, it is possible to flexibly respond to the change of the Web page location.

[0058] Furthermore, although the model information of the printer is also used to search the driver in the second and third embodiments described above, this is optional. For example, in the second embodiment, if the databases are provided for the respective printer models in the server SV and the printer PT is set to acquire its own database in advance, it is possible to search the driver corresponding to the OS of the client PC without using the model information of the printer. In the third embodiment, if the database query modules that differ by printer model are provided in the server SV and the printer PT specifies its own database query module, it is possible to search the driver corresponding to the OS of the client PC without using the model information of the printer.